How to be Firmly Antifounded?

Baltic fun in progress

Tarmo Uustalu, Tallinn

Joint work with
Venanzio Capretta, Nijmegen
Varmo Vene, Tartu

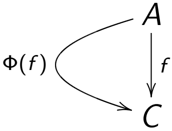
Fun in the Afternoon, Cambridge, 17 May 2007

Antifounded?

- *Recursive coalgebras* are a simple formulation of *wellfounded* ($\approx$ terminating) structured general recursion for the context of *total* functional programming.
- Why then not have some *dualization* fun with *antifounded* ($\approx$ productive) structured general recursion, i.e., look at *iterative algebras*?
- This is fun indeed and is not without surprises as **Set** is far from self-dual.
- This talk: We study a basic sufficient condition of iterativeness.

Unstructured general recursion

- Unstructured recursive function specification:



$\Phi(f)$ – recursive definiens

- We are in **Set** (or in some other category of total functions) and look for unique solutions (not for canonical ones among multiple solutions) to recursive specifications.
- So we ask: What are some useful ways to prove that a general recursive *specification* is a *definition* (i.e., a definite description)?
- This one is a definition, but how to learn it is?

$$
\begin{aligned}
&\text{qsort} : \text{ListNat} \to \text{ListNat} \\
&\text{qsort}\,([]) =_{\text{df}} [] \\
&\text{qsort}\,(n : ns) =_{\text{df}} \text{qsort}\,(ns_{\leq n}) ++ (n : \text{qsort}\,(ns_{>n}))
\end{aligned}
$$

Structured general recursion

- Structured recursive specification:

$$\begin{array}{ccc} FA & \xleftarrow{\ \alpha\ } & A \\ {\scriptstyle Ff}\big\downarrow & & \big\downarrow{\scriptstyle f} \\ FC & \xrightarrow[\ \gamma\ ]{} & C \end{array}$$

F – branching type of recursive call trees (an endofunctor on a category $\mathcal{C}$ of types and functions, e.g., **Set**)

α – marshalling of arguments for recursive calls (an F-coalgebra)

Ff – recursive calls (functorial action of F on f)

γ – collection of recursive call results (an F-algebra)

- f is specified as an *F-coalgebra-to-algebra map* from (A, α) to (C, γ).

- Quicksort specification has this structure:

$$
\begin{array}{ccc}
1 + \mathsf{Nat} \times \mathsf{List\,Nat} \times \mathsf{List\,Nat} & \xleftarrow{\mathsf{qsplit}} & \mathsf{List\,Nat} \\
{\scriptstyle 1+\mathsf{Nat}\times\mathsf{qsort}\times\mathsf{qsort}}\big\downarrow & & \big\downarrow{\scriptstyle \mathsf{qsort}} \\
1 + \mathsf{Nat} \times \mathsf{List\,Nat} \times \mathsf{List\,Nat} & \xrightarrow[\mathsf{qmerge}]{} & \mathsf{List\,Nat}
\end{array}
$$

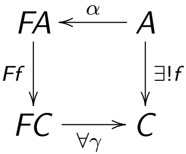
$$
\begin{aligned}
\mathsf{qsplit}\,([]) &=_{\mathrm{df}} \mathsf{inl}(*) \\
\mathsf{qsplit}\,(n : ns) &=_{\mathrm{df}} \mathsf{inr}(n, ns_{\leq n}, ns_{>n})
\end{aligned}
$$

$$
\begin{aligned}
\mathsf{qmerge}\,(\mathsf{inl}(*)) &=_{\mathrm{df}} [] \\
\mathsf{qmerge}\,(\mathsf{inr}(n, ns_0, ns_1)) &=_{\mathrm{df}} ns_0 +\!\!+ (n : ns_1)
\end{aligned}
$$

- Nested calls etc. do not fit into this structured framework.

Recursive coalgebras

- Given an endofunctor F on a category $\mathcal{C}$, an F-coalgebra $(A, \alpha : A \to FA)$ is *recursive* if for any F-algebra $(C, \gamma : FC \to C)$ there is a unique coalgebra-to-algebra map, i.e., if it satisfies



- Having the coalgebra recursive is wellfounded general recursion, a way to ensure that a recursive spec uniquely determines a function irrespective of the algebra.
- In quicksort, qsplit is recursive: we could replace qmerge with any other algebra, the spec would still be a definition.

Examples

- These coalgebras are recursive:
  - $FC =_{\mathrm{df}} 1 + C$, $A =_{\mathrm{df}} \mathsf{Nat}$, $\alpha : \mathsf{Nat} \to 1 + \mathsf{Nat}$, $\alpha(\mathsf{o}) =_{\mathrm{df}} \mathsf{inl}(\star)$, $\alpha(\mathsf{s}(n)) =_{\mathrm{df}} \mathsf{inr}(n)$
    – naturals with their inverse-of-initial-algebra coalgebra structure (predecessor)
  - $FC =_{\mathrm{df}} 1 + C$, $A =_{\mathrm{df}} \mathsf{Nat}_{<13}$, $\alpha : \mathsf{Nat}_{<13} \to 1 + \mathsf{Nat}_{<13}$, $\alpha(\mathsf{o}) =_{\mathrm{df}} \mathsf{inl}(\star)$, $\alpha(\mathsf{s}(n)) =_{\mathrm{df}} \mathsf{inr}(n)$
    – naturals smaller than 13
  - $FC =_{\mathrm{df}} 1 + C$, $A =_{\mathrm{df}} \mathsf{Nat}$, $\alpha : \mathsf{Nat} \to 1 + \mathsf{Nat}$, $\alpha(\mathsf{o}) =_{\mathrm{df}} \mathsf{inl}(\star)$, $\alpha(\mathsf{s}(\mathsf{o})) =_{\mathrm{df}} \mathsf{inl}(*)$, $\alpha(\mathsf{s}(\mathsf{s}(n))) =_{\mathrm{df}} \mathsf{inr}(n)$
    – naturals with subtraction of 2

- These are recursive too:
  - $FC =_{\rm df} 1 + C \times (1 \times C)$, $A =_{\rm df} \mathsf{Nat}$,
    $\alpha : \mathsf{Nat} \to 1 + \mathsf{Nat} \times (1 + \mathsf{Nat})$, $\alpha(\mathsf{o}) =_{\rm df} \mathsf{inl}(\star)$,
    $\alpha(\mathsf{s}(\mathsf{o})) =_{\rm df} \mathsf{inr}(\mathsf{o}, \mathsf{inl}(*))$, $\alpha(\mathsf{s}(\mathsf{s}(n))) =_{\rm df} \mathsf{inr}(\mathsf{s}(n), \mathsf{inr}(n))$
    – naturals with subtraction of 1 and 2
  - $FC =_{\rm df} 1 + C$, $A =_{\rm df} \mathsf{Nat}$, $\alpha : \mathsf{Nat} \to 1 + \mathsf{Nat}$,
    $\alpha(\mathsf{o}) =_{\rm df} \mathsf{inl}(\star)$, $\alpha(\mathsf{s}(n)) =_{\rm df} \mathsf{inr}(\mathsf{halve}(n))$
    – naturals with predecessor division by 2
- These are not recursive:
  - $FC =_{\rm df} 1 + C$, $A =_{\rm df} \mathsf{Nat}^\infty$, $\alpha : \mathsf{Nat}^\infty \to 1 + \mathsf{Nat}^\infty$,
    $\alpha(\mathsf{o}^\infty) =_{\rm df} \mathsf{inl}(\star)$, $\alpha(\mathsf{s}^\infty(n)) =_{\rm df} \mathsf{inr}(n)$
    – conaturals (naturals plus an infinite number) with their final coalgebra structure (predecessor)
  - $FC =_{\rm df} C$, A any non-empty set, $\alpha : A \to A$, $\alpha(x) = x$
    – a non-empty set with its identity
    (the coalgebra for recursive equations $f(x) = \gamma(f(x))$)

The dual: Iterative algebras

- An F-algebra $(C, \gamma : FC \to C)$ is *iterative* if for any F-coalgebra $(A, \alpha : A \to FA)$ there is a unique coalgebra-to-algebra map.

$$\begin{array}{ccc} FA & \xleftarrow{\forall \alpha} & A \\ {\scriptstyle Ff}\downarrow & & \downarrow{\scriptstyle \exists! f} \\ FC & \xrightarrow[\gamma]{} & C \end{array}$$

- This is antifounded recursion. The recursive spec uniquely determines a function irrespective of the coalgebra.

Examples

- These are iterative:
  - $FA =_{\mathrm{df}} E \times A$, $C =_{\mathrm{df}} \mathsf{Str} E$, $\gamma : E \times \mathsf{Str} E \to \mathsf{Str} E$, $\gamma(e, es) =_{\mathrm{df}} e : es$
    – streams with the inverse-of-final-coalgebra algebra structure (cons)
  - $FA =_{\mathrm{df}} E \times E \times A$, $C =_{\mathrm{df}} \mathsf{Str} E$, $\gamma : E \times E \times \mathsf{Str} E \to \mathsf{Str} E$, $\gamma(e_0, e_1, es) =_{\mathrm{df}} e_0 : e_1 : es$
    – streams and consing two elements to a stream at once
  - $FA =_{\mathrm{df}} E \times A$, $C =_{\mathrm{df}} \mathsf{Str} E$, $\gamma : E \times \mathsf{Str} E \to \mathsf{Str} E$, $\gamma(e_0, es) =_{\mathrm{df}} e_0 : e_0 : es$
    – streams and consing two copies of an element to a stream

- But the following are iterative too:
  - $FA =_{\mathrm{df}} E \times A \times A$, $C =_{\mathrm{df}} \mathsf{Str}E$,
    $\gamma : E \times \mathsf{Str}E \times \mathsf{Str}E \to \mathsf{Str}E$,
    $\gamma(e, es_0, es_1) =_{\mathrm{df}} e : \mathsf{merge}(es_0, es_1)$
    – streams and consing an element to the interleaving of two streams
  - $FA =_{\mathrm{df}} E \times A$, $C =_{\mathrm{df}} \mathsf{Str}E$, $\gamma : E \times \mathsf{Str}E \to \mathsf{Str}E$,
    $\gamma(e, es) =_{\mathrm{df}} e : \mathsf{double}(es)$
    – streams and consing an element to the doubling of a stream
- This one isn't:
  - $FA =_{\mathrm{df}} 1 + A$, $C =_{\mathrm{df}} \mathsf{Nat}$, $\alpha : 1 + \mathsf{Nat} \to \mathsf{Nat}$,
    $\alpha(\mathsf{inl}(\star)) =_{\mathrm{df}} \mathsf{o}$, $\alpha(\mathsf{inr}(n)) =_{\mathrm{df}} \mathsf{s}(n)$
    – naturals with their initial-algebra structure (zero and successor)

Simple sufficient conditions for recursiveness (and iterativeness)

- There are various sufficient conditions for recursiveness.
- Some are assume little about the category and functor, but are about specific kinds of coalgebras and therefore of limited applicability.
  - For any functor F on any category $\mathcal{C}$, if F has an initial algebra $(\mu F, \mathsf{in}_F)$, then $(\mu F, \mathsf{in}_F^{-1})$ is a recursive F-coalgebra (in fact the final one).
  - For any functor F on any category $\mathcal{C}$, if (A, α) is a recursive F-coalgebra, then so is $(FA, F\alpha)$.
  - For any functor F on any category $\mathcal{C}$, if (A, α) is a recursive F-coalgebra, then $(A, F\alpha \circ \alpha)$ is a recursive FF-coalgebra.
- Such conditions dualize straightforwardly without losing their usefulness.

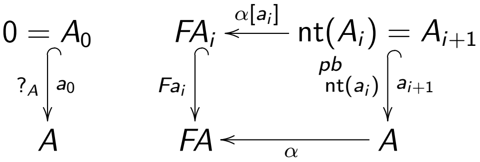
A deeper condition

- Here is a condition applicable to any coalgebra, but assuming more about the category and the functor.
- Assume $\mathcal{C}$ has pullbacks, an initial object and colimits of ω-chains of monos and F preserves monos. (This holds for **Set** and many **Set** functors.)
- Given an F-coalgebra (A, α), define the "nexttime" version of a subset (U, u) of A (i.e., a mono $u : U \hookrightarrow A$) as the pullback

$$
\begin{array}{ccc}
FU & \xleftarrow{\ \alpha[u]\ } & \mathrm{nt}(U) \\
{\scriptstyle Fu}\big\downarrow & pb & \big\downarrow{\scriptstyle \mathrm{nt}(u)} \\
FA & \xleftarrow[\ \alpha\ ]{} & A
\end{array}
$$

Set-theoretically $\mathrm{nt}(U) = \{a \in A \mid \alpha(a) \in FU\}$.

- Define an ω-sequence of subobjects $(A_i, a_i)_i$ of A by

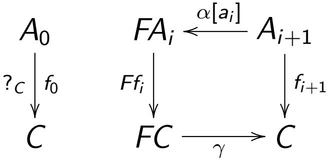


- The objects A_i form a chain $(e_i : A_i \hookrightarrow A_{i+1})_i$ (def. omitted).
- Let

$$A^* =_{\mathrm{df}} \mathrm{colim}_i A_i$$

- Set-theoretically $A^* \cong \bigcup_i A_i$. (So in particular $A^* \subseteq A$).
- Now: If $A^* \cong A$, then (A, α) is recursive.

- Specifically, for any F-algebra (C, γ), the unique coalgebra-to-algebra map $f : A \to C$ is given by the unique mediating map $A^* \to C$ for the cocone $(f_i : A_i \to C)_i$ defined by



Examples

- $FC =_{\mathrm{df}} 1 + C$, $A =_{\mathrm{df}} \mathsf{Nat}$, $\alpha : \mathsf{Nat} \to 1 + \mathsf{Nat}$,
  $\alpha(\mathsf{o}) =_{\mathrm{df}} \mathsf{inl}(\star)$, $\alpha(\mathsf{s}(n)) =_{\mathrm{df}} \mathsf{inr}(n)$
  – naturals with predecessor
  $A_0 = 0$, $A_1 = \{0\}$, $A_2 = \{0, 1\}$, ...,
  $A_i = \mathsf{Nat}_{<i}$,
  $A^* = \mathsf{Nat} = A$ – recursive!
- $FC =_{\mathrm{df}} 1 + C$, $A =_{\mathrm{df}} \mathsf{Nat}_{<13}$, $\alpha : \mathsf{Nat}_{<13} \to 1 + \mathsf{Nat}_{<13}$,
  $\alpha(\mathsf{o}) =_{\mathrm{df}} \mathsf{inl}(\star)$, $\alpha(\mathsf{s}(n)) =_{\mathrm{df}} \mathsf{inr}(n)$
  – naturals smaller than 13
  $A_0 = 0$, $A_1 = \{0\}$, $A_2 = \{0, 1\}$, ...,
  $A_i = \mathsf{Nat}_{<i}$ for $i < 13$, $A_i = \mathsf{Nat}_{<13}$ for $i \geq 13$,
  $A^* = \mathsf{Nat}_{<13} = A$ – recursive!

- $FC =_{\mathrm{df}} 1 + C$, $A =_{\mathrm{df}} \mathsf{Nat}$, $\alpha : \mathsf{Nat} \to 1 + \mathsf{Nat}$,
  $\alpha(\mathsf{o}) =_{\mathrm{df}} \mathsf{inl}(\star)$, $\alpha(\mathsf{s}(\mathsf{o})) =_{\mathrm{df}} \mathsf{inl}(*)$, $\alpha(\mathsf{s}(\mathsf{s}(n))) =_{\mathrm{df}} \mathsf{inr}(n)$
  – naturals with subtraction of 2
  $A_0 = 0$, $A_1 = \{0, 1\}$, $A_2 = \{0, 1, 2, 3\}$, ...,
  $A_i = \mathsf{Nat}_{\leq 2i-1}$,
  $A^* = \mathsf{Nat} = A$ – recursive!
- $FC =_{\mathrm{df}} 1 + C$, $A =_{\mathrm{df}} \mathsf{Nat}$, $\alpha : \mathsf{Nat} \to 1 + \mathsf{Nat}$,
  $\alpha(\mathsf{o}) =_{\mathrm{df}} \mathsf{inl}(\star)$, $\alpha(\mathsf{s}(n)) =_{\mathrm{df}} \mathsf{inr}(\mathsf{halve}(n))$
  – naturals with predecessor division by 2
  $A_0 = 0$, $A_1 = \{0\}$, $A_2 = \{0, 1, 2\}$,
  $A_3 = \{0, 1, 2, 3, 4, 5, 6\}$, $A_4 = \{0..14\}$, ...,
  $A_i = \mathsf{Nat}_{<2^i-2}$,
  $A^* = \mathsf{Nat} = A$ – recursive!

- $FC =_{\mathrm{df}} 1 + C$, $A =_{\mathrm{df}} \mathsf{Nat}^\infty$, $\alpha : \mathsf{Nat}^\infty \to 1 + \mathsf{Nat}^\infty$,
  $\alpha(\mathsf{o}^\infty) =_{\mathrm{df}} \mathsf{inl}(\star)$, $\alpha(\mathsf{s}^\infty(n)) =_{\mathrm{df}} \mathsf{inr}(n)$
  – conaturals with predecessor
  $A_0 = 0$, $A_1 = \{0^\infty\}$, $A_2 = \{0^\infty, 1^\infty\}$, ...,
  $A_i = \mathsf{Nat}^\infty{}_{< i^\infty}$,
  $A^* = \mathsf{Nat}^\infty \setminus \{\infty\} \not\cong \mathsf{Nat}^\infty = A$ – not recursive!
- $FC =_{\mathrm{df}} C$, A any non-empty set, $\alpha : A \to A$, $\alpha(a) = a$
  – non-empty set with identity
  $A_0 = 0$, $A_1 = 0$, ..., $A_i = 0$, $A^* = 0 \not\cong A$ – not recursive!

- Notably this condition of recursiveness does not assume ω-cocontinuity of F.
- Why is it not needed? We are nowhere trying to construct the least fixpoint of F – recursive coalgebras are not least fixpoints of functors, differently from initial algebras. Also the condition is only sufficient, not necessary.
- In fact, we are instead approximating the least fixpoint of the nt operation on subsets of A. For a sufficient condition, it need not be reached by ω or exist at all.
- In category theory literature, variants of this condition appear in the works by P. Taylor and by J. Adámek et al.
- In type theory, it is known as the Bove–Capretta method.

The dual condition

- This condition dualizes, but in **Set** we have to be careful about simplifications.
- Assume $\mathcal{C}$ has pushouts, a final object and limits of ω-opchains of epis and F preserves epis.
- Given an F-algebra (C, γ), define the "nexttime" version of a quotient (Q, q) of C (i.e., an epi $q : C \twoheadrightarrow Q$) as the pushout

$$\begin{array}{ccc} FC & \xrightarrow{\gamma\langle q\rangle} & C \\ {\scriptstyle Fq}\downarrow & \quad po & \downarrow{\scriptstyle \mathrm{nt}(q)} \\ FQ & \xrightarrow[\gamma]{} & \mathrm{nt}(Q) \end{array}$$

Set-theoretically $\mathrm{nt}(C/{\sim}) = C/\mathrm{nt}(\sim)$ where

$$\mathrm{nt}(\sim) =_{\mathrm{df}} \{(\gamma(t), \gamma(u) \mid (t, u) \in FC \times FC, (t, u) \in F(\sim)\}^*$$

- Define an ω-sequence of quotients (C_i, c_i) of C by

$$
\begin{array}{ccccc}
C & \qquad & FC & \xrightarrow{\gamma\langle c_i \rangle} & C \\
{\scriptstyle !_C}\big\downarrow{\scriptstyle c_0} & & {\scriptstyle Fc_i}\big\downarrow & {\scriptstyle po} & {\scriptstyle \mathrm{nt}(c_i)}\big\downarrow{\scriptstyle c_{i+1}} \\
1 = C_0 & & FC_i & \xrightarrow[\gamma]{} & \mathrm{nt}(C_i) = C_{i+1}
\end{array}
$$

- The objects C_i form an opchain $(p_i : C_{i+1} \twoheadrightarrow C_i)_i$ (def. omitted).

- Let

$$C^* =_{\mathrm{df}} \lim_i C_i$$

- Set-theoretically

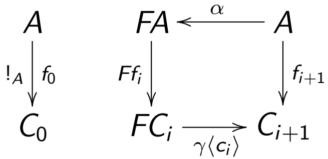
$$C^* \cong \{(c_i(x_i))_i | x_i \in C, p_i(c_{i+1}(x_{i+1})) = c_i(x_i)\}$$

- This is not the same as

$$\begin{aligned} C/(\cap_i \sim_i) &\cong \{(c_i(x))_i | x \in C\} \\ &= \{(c_i(x))_i | x \in C, p_i(c_{i+1}(x)) = c_i(x)\} \end{aligned}$$

- Moreover, C^* is in general *not* a quotient of C.
  In such cases $C^* = \lim_i C_i$ is not a limit of $(C_i, c_i)_i$ in the category of quotients of C and in fact this opchain has no limit there.
- Now, again: If $C^* \cong C$, then (C, γ) is iterative.

- Here, for any F-coalgebra (A, α), the unique coalgebra-to-algebra map $f : A \to C$ is given by the unique mediating map $A \to C^*$ for the cone $(f_i : A \to C_i)_i$ defined by



Examples

- $FA =_{\mathrm{df}} E \times A$, $C =_{\mathrm{df}} \mathsf{Str}E$, $\gamma : E \times \mathsf{Str}E \to \mathsf{Str}E$, $\gamma(e, es) =_{\mathrm{df}} e : es$
  – streams with the inverse-of-final-coalgebra algebra structure (cons)

$$
\begin{array}{rcl}
C_0 & = & \{\{es \mid es \in \mathsf{Str}E\}\} \cong 1, \\
C_1 & = & \{\{e_0 : es \mid es \in \mathsf{Str}E\} \mid e_0 \in E\}, \\
C_2 & = & \{\{e_0 : e_1 : es \mid es \in \mathsf{Str}E\} \mid e_0, e_1 \in E\}, \\
 & \ldots & \\
C_i & = & \{\{e_0 : \ldots : e_{i-1} : es \mid es \in \mathsf{Str}E\} \mid e_0, \ldots, e_{i-1} \in E\}, \\
C^* & = & \mathsf{Str}E = C \text{ -- iterative}
\end{array}
$$

- $FA =_{\mathrm{df}} E \times E \times A$, $C =_{\mathrm{df}} \mathsf{Str} E$,
  $\gamma : E \times E \times \mathsf{Str} E \to \mathsf{Str} E$, $\gamma(e_0, e_1, es) =_{\mathrm{df}} e_0 : e_1 : es$
  – streams and consing two elements to a stream at once

$$
\begin{array}{rcl}
C_0 & = & \{\{es \mid es \in \mathsf{Str}E\}\} \cong 1, \\
C_1 & = & \{\{e_0 : e_1 : es \mid es \in \mathsf{Str}E\} \mid e_0, e_1 \in E\}, \\
C_2 & = & \{\{e_0 : \ldots : e_3 : es \mid es \in \mathsf{Str}E\} \mid e_0, \ldots, e_3 \in E\}, \\
& \ldots & \\
C_i & = & \{\{e_0 : \ldots : e_{2i-1} : es \mid es \in \mathsf{Str}E\} \mid e_0, \ldots, e_{2i-1} \in E\} \\
C^* & = & \mathsf{Str}E = C\text{– iterative}
\end{array}
$$

- $FA =_{\rm df} E \times A$, $C =_{\rm df} \mathsf{Str}E$, $\gamma : E \times \mathsf{Str}E \to \mathsf{Str}E$, $\gamma(e_0, es) =_{\rm df} e_0 : e_0 : es$
  – streams and consing two copies of an element to a stream

$$
\begin{array}{rcl}
C_0 & = & \{\{es \mid es \in \mathsf{Str}E\}\} \cong 1, \\
C_1 & = & \{\{e_0 : e_0 : es \mid es \in \mathsf{Str}E\} \mid e_0 \in E\} \\
 & \cup & \{\{e_0 : e_0' : es\} | e_0, e_0' \in E, e_0 \neq e_0', es \in \mathsf{Str}E\}, \\
C_2 & = & \{\{e_0 : e_0 : e_1 : e_1 : es \mid es \in \mathsf{Str}E\} \mid e_0, e_1 \in E\} \\
 & \cup & \{\{e_0 : e_0' : e_1 : e_1' : es\} | \\
 & & \quad e_0, e_0', e_1, e_1' \in E, e_0 \neq e_0' \vee e_1 \neq e_1', es \in \mathsf{Str}E\}, \\
 & \ldots & \\
C^* & = & \mathsf{Str}E = C\text{– iterative}
\end{array}
$$

- $FA =_{\mathrm{df}} E \times A \times A$, $C =_{\mathrm{df}} \mathrm{Str}E$,
  $\gamma : E \times \mathrm{Str}E \times \mathrm{Str}E \to \mathrm{Str}E$,
  $\gamma(e, es_0, es_1) =_{\mathrm{df}} e : \mathrm{merge}(es_0, es_1)$
  – streams and consing an element to the interleaving of two streams

$$
\begin{aligned}
C_0 &= \{\{es \mid es \in \mathrm{Str}E\}\} \cong 1, \\
C_1 &= \{\{e_0 : es \mid es \in \mathrm{Str}E\} \mid e_0 \in E\}, \\
C_2 &= \{\{e_0 : e_1 : e_2 : es \mid es \in \mathrm{Str}E\} \mid e_0, e_1, e_2 \in E\}, \\
C_3 &= \{\{e_0 : \ldots : e_6 : es \mid es \in \mathrm{Str}E\} \mid e_0, \ldots e_6 \in E\}, \\
&\ldots \\
C_i &= \{\{e_0 : \ldots : e_{2^i-2} : es \mid es \in \mathrm{Str}E\} \mid \\
&\qquad e_0, \ldots, e_{2^i-2} \in E\}, \\
C^* &= \mathrm{Str}E = C\text{– iterative}
\end{aligned}
$$

- $FA =_{\mathrm{df}} E \times A$, $C =_{\mathrm{df}} \mathsf{Str}E$, $\gamma : E \times \mathsf{Str}E \to \mathsf{Str}E$, $\gamma(e, es) =_{\mathrm{df}} e : \mathsf{double}(es)$
  – streams and consing an element to the doubling of a stream

$$
\begin{array}{rcl}
C_0 & = & \{\{es \mid es \in \mathsf{Str}E\}\} \cong 1, \\
C_1 & = & \{\{e_0 : es \mid es \in \mathsf{Str}E\} \mid e_0 \in E\}, \\
C_2 & = & \{\{e_0 : e_1 : e_1 : es \mid es \in \mathsf{Str}E\} \mid e_0, e_1 \in E\} \\
 & \cup & \{\{e_0 : e_{10} : e_{11} : es\} \mid \\
 & & \quad e_0, e_{10}, e_{11} \in E, \neg e_{10} = e_{11}, es \in \mathsf{Str}E\} , \\
C_3 & = & \{\{e_0 : e_1 : e_1 : e_2 : e_2 : e_2 : e_2 : es \mid es \in \mathsf{Str}E\} \mid \\
 & & \quad e_0, e_1, e_2 \in E\} \\
 & \cup & \{\{e_0 : e_{10} : e_{11} : e_{20} : e_{21} : e_{22} : e_{23} : es\} \mid \\
 & & \quad e_0, e_{10}, e_{11}, e_{20}, e_{21}, e_{22}, e_{23} \in E, \\
 & & \quad \neg e_{10} = e_{11} \vee \neg e_{20} = e_{21} = e_{22} = e_{23}, es \in \mathsf{Str}E\} \\
C^* & = & \mathsf{Str}E = C\text{– iterative}
\end{array}
$$

- $FA =_{\mathrm{df}} 1 + A$, $C =_{\mathrm{df}} \mathsf{Nat}$, $\alpha : 1 + \mathsf{Nat} \rightarrow \mathsf{Nat}$, $\alpha(\mathsf{inl}(\star) =_{\mathrm{df}} \mathsf{o}$, $\alpha(\mathsf{inr}(n)) =_{\mathrm{df}} \mathsf{s}(n)$
  – naturals with their initial-algebra structure (zero and successor)

$$\begin{array}{rcl} C_0 & = & \{\mathsf{Nat}\} \cong 1, \\ C_1 & = & \{\{0\}, \mathsf{Nat}_{\geq 1}\}, \\ C_2 & = & \{\{0\}, \{1\}, \mathsf{Nat}_{\geq 2}\}, \\ & \ldots & \\ C_i & = & \{\{0\}, \ldots, \{i\}, \mathsf{Nat}_{\geq i}\}, \\ C^* & \cong & \{\{i\} \mid i \in \mathsf{Nat}\} \cup \{\infty\} \cong \mathsf{Nat}^\infty \not\cong \mathsf{Nat} = C \end{array}$$

– not iterative

Note that $C / \cap_i \sim_i = \mathsf{Nat} = C$, so it is important to calculate C^* correctly!

Antifounded general recursion is fun, isn't it?